

TP 1 : Outils et traitements de base sous Matlab

Ce premier TP a pour but de vous familiariser avec le logiciel Matlab et les différents outils de base pour le traitement des images disponibles dans la bibliothèque Matlab *Image processing Toolbox*. Vous êtes invités à consulter régulièrement la documentation des différentes fonctions Matlab pour comprendre les opérations qu'elles effectuent :

```
>> help function    % Pour une description succincte
>> doc function    % Pour une description plus détaillée et des exemples
```

Quelques conseils avant de commencer

- On rappelle qu'un compte-rendu de TP comporte une introduction et une conclusion et que cette conclusion ne doit pas être la copie de l'introduction au passé.
- On rappelle également que les figures doivent comporter un titre, et que, le cas échéant, les axes des graphes doivent avoir un nom (et une unité quand elle est connue).
- Créer un script Matlab *TP.m*, ne faites pas tout le TP en lignes de commandes.
- Dans le compte rendu, on doit retrouver les figures qui vous paraissent pertinentes (parfois un schéma suffit pour expliquer un phénomène observé sur une série d'image) et ne pas mettre le code dans le compte-rendu sauf si cela est explicitement demandé.

1 Manipulation d'images sous Matlab (2h)

Cette partie n'a pas besoin de figurer sur le compte-rendu.

1.1 Matrices

Matlab est un langage de programmation dédié au calcul numérique, et en particulier au calcul matriciel (Matlab = *Matrix Laboratory*).

Pour la création d'images de synthèse, on peut avoir besoin des fonctions suivantes :

```
>> A = zeros(M,N);    % Création d'une matrice nulle de taille  $M \times N$ .
>> B = ones(M,N);    % Création d'une matrice de taille  $M \times N$  remplie de 1.
```

Voici une liste de fonctions utiles pour traiter les matrices, à vous d'aller voir dans l'aide ce qu'elles font : `min`, `max`, `mean`, `median`, `sum`, `size`.

L'élément (i,j) d'une matrice M est identifié par `M(i,j)`. Si l'une des deux coordonnées est remplacée par l'opérateur `:`, alors c'est toute une ligne ou toute une colonne de la matrice qui est modifiée. Si l'on souhaite modifier seulement les valeurs comprises entre les indices `i1` et `i2`, on utilisera l'opérateur `i1 : i2`. Si cette fois-ci on veut modifier une valeur tous les k échantillons entre `i1` et `i2` il suffit d'utiliser l'opérateur `i1 : k : i2`. Par défaut quand on utilise l'opérateur `i1 : i2`, implicitement le pas k est égal à 1, si on précise la valeur du pas, Matlab comptera de `i1` à `i2` avec un pas égal à la valeur k .

Manipulation 1

- ➡ Créer une matrice de taille 9×9 remplie de valeurs 255.
- ➡ Modifier la matrice pour obtenir la matrice suivante :

$$\begin{pmatrix} 255 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 44 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

1.2 Images

Puisqu'une image est stockée par une matrice, les modifications de matrice que l'on a vu précédemment peuvent être appliquées de la même façon à une image. Et inversement, on peut utiliser les outils utilisés pour manipuler des images (affichage, enregistrement, etc) sur des matrices.

1.2.1 Manipulation des images

Manipulation 2

- ➡ Charger (`imread`) et afficher (`imshow`) l'image 'mandrill_color.tif'
 - ➡ Afficher chacune des composantes rouge, verte et bleue de l'image.
 - ➡ Convertir l'image 'mandrill_color.tif' en niveaux de gris avec la fonction `rgb2gray`.
 - ➡ Visualiser l'image convertie en niveaux de gris en précisant dans `imshow` la plage d'intensité. Tester successivement `[0,1]`, `[0,100]`, `[100,255]` et `[]`. Il est possible d'afficher la barre de couleur correspondante à l'image affichée avec l'option `colorbar()`.
 - ➡ Sous-échantillonner l'image convertie en niveaux de gris avec un pas $k = 2, 3, 5, 10$ et visualiser l'image.
 - ➡ Que se passe-t-il si l'on multiplie l'image convertie en niveaux de gris par une matrice remplie de 1 ?
-

La conversion d'une image I dont les intensités sont codés en `uint8` vers une image dont les intensités sont codés en `double` se fait à l'aide de la fonction `double(I)`. On fera attention à l'affichage des images codées en `double` avec la fonction `imshow`, il faut penser à utiliser la bonne option.

1.2.2 Affichage d'une image de synthèse

On rappelle qu'une image couleur est une image à trois bandes (R,G,B). C'est donc une matrice à 3 dimensions : les deux premières sont les dimensions de l'image, la troisième correspond aux bandes R,G,B. On souhaite créer l'image ci-dessous.

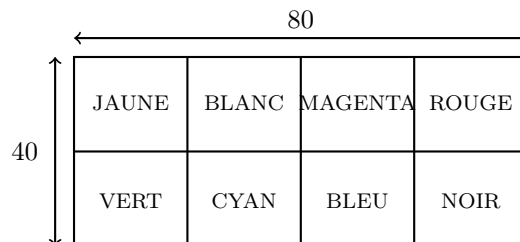


Image couleur à synthétiser

Manipulation 3

- ➡ Synthétiser et afficher l'image 40×80 schématisée ci-dessus.
 - ➡ Enregistrer l'image obtenue avec `imwrite`.
-

Une manière simple de réaliser cet exercice est tout d'abord de créer une image noire (tableau rempli de zéros) aux bonnes dimensions, puis d'affecter la valeur souhaitée aux éléments de la matrice correspondant.

```
>> I(i1:i2, j1:j2,b) = k    % met à la valeur k tous les pixels de la bande b
de l'image I compris entre les lignes i1 et i2 et les colonnes j1 et j2.
>> imwrite(I, 'image.jpg') % enregistrement de l'image au format jpeg.
```

2 Transformée de Fourier bidimensionnelle (1h)

La transformée de Fourier d'une image I de taille $M \times N$ est définie par :

$$\mathcal{I}(u, v) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i, j) e^{-j 2\pi \left(\frac{ui}{M} + \frac{vj}{N} \right)}$$

Vous trouverez ci-dessous quelques fonctions utiles pour calculer la transformée de Fourier bidimensionnelle.

```
>> TF = 1/(M*N)*fft2(I);
>> TF = fftshift(TF);
>> module = abs(TF);
>> phase = angle(TF)
```

⚡ Pourquoi doit-on normaliser la transformée de Fourier calculée par Matlab par le nombre de pixels dans l'image (indication : regarder dans la documentation de Matlab comment est implémentée la fonction `fft2`).

⚡ Que fait la fonction `fftshift` ? Pourquoi est-ce utile de faire appel à cette fonction avant d'afficher le module et la phase de la transformée de Fourier ?

Manipulation 4

- ➡ Charger l'image 'briques.jpg' puis afficher l'image.
 - ➡ Convertir l'image en niveaux de gris.
 - ➡ Calculer sa transformée de Fourier 2D discrète et afficher son module.
-

⚡ Comment interpréter le module de la transformée de Fourier de l'image 'briques.jpg' ?

Manipulation 5

- ➡ Charger l'image 'briques_rotation.jpg' puis afficher l'image.
 - ➡ Convertir l'image en niveaux de gris.
 - ➡ Calculer sa transformée de Fourier 2D discrète et afficher son module.
-

⚡ Comment interpréter le module de la transformée de Fourier de l'image 'briques_rotation.jpg' ? Quelle différence avec l'image 'briques.jpg' sans rotation ?

3 Convolution (1h)

La convolution de deux images I_1 et I_2 sous Matlab est réalisée par la fonction `conv2` qui prend en arguments les deux images I_1 et I_2 et une `option` ('full', 'same' ou 'valid') indiquant la taille de l'image convoluée.

```
>> Iconv = conv2(I_1, I_2, option);
```

On définit le filtre H par :

$$H = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Manipulation 6

- ➡ Charger et afficher l'image 'Fig0327(a)(tungsten_original).tif'
 - ➡ Créer le filtre H décrit précédemment.
 - ➡ Convoluer cette image avec le filtre H .
 - ➡ Afficher le résultat et comparer à l'image originale. Attention à l'encodage des intensités sur ces deux images. Les opérations matricielles simples (+, -, .*, ./) nécessitent que les valeurs dans les deux matrices aient le même type.
-

✎ Commenter l'influence du filtre H sur l'image? Analyser le résultat. Une attention particulière sera apportée aux valeurs d'intensité de l'image après convolution.